

# C#프로그래밍 프로젝트 결과 보고서

18013189 컴퓨터공학과 차윤범

## 1. 개요

### - 프로젝트 소개

그림판 프로그램으로 pictureBox안에서 그림을 그리는 구조입니다.

그림을 그리는 기능은 선, 사각형, 원 3가지 기능으로 구성하였고 프로젝트 제안서 내용에 있는 자유곡선 기능은 구현하는데에 실패하였습니다.

자유곡선 기능 구현에 실패하는 대신 List 컬렉션을 이용하여 그림을 그릴 때마다 List 컬렉션에 Add, RemoveAt 함수를 사용하여 undo, redo 기능을 구현했습니다.

레이아웃에 있는 버튼 이벤트로 동작도 가능하게 하고, menustrip 도구를 이용해서도 선, 사각형, 원을 그릴 수 있게 구현했습니다.

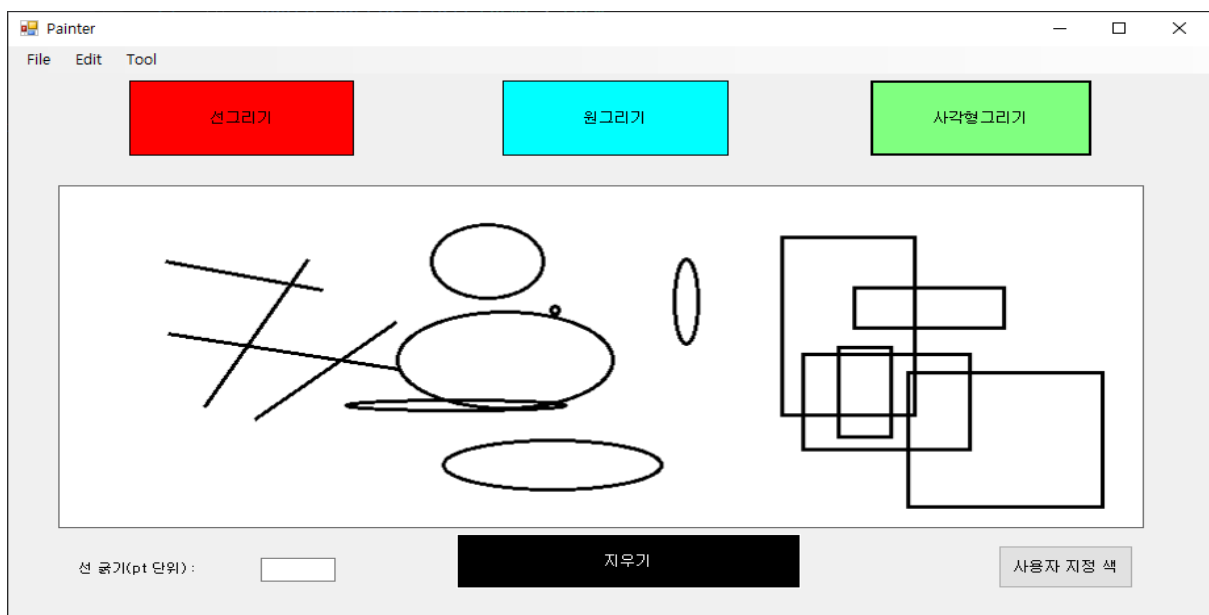
그리고 그리는 모양들의 굵기와 색을 지정하여 커스텀을 할 수 있게 기능을 구현했습니다.

파일을 불러오고 저장할 수 있는 기능을 구현했습니다.

## 2. 개발 내용

### 최종 구현

### - 레이아웃



## - 기능

빨간 버튼을 누르면 pictureBox1에 직선을 그릴 수 있는 모습을 확인할 수 있고 cyan 색의 버튼을 누르면 원을 그릴 수 있는 기능, 그 오른쪽 연두색 버튼을 누르면 사각형을 그릴 수 있는 기능을 확인할 수 있습니다.

```
//선 굵기 지정
참조 1개
private void textBox1_TextChanged(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        width = Convert.ToInt32("0");
        pn.Width = width;
    }
    else
    {
        width = Convert.ToInt32(textBox1.Text);
        pn.Width = width;
    }
}
```

좌측 아래에 선 굵기를 지정할 수 있습니다. 선 굵기를 textbox에 입력하고 따로 enter없이 바로 식별하여 굵기를 줄이고 늘릴 수 있습니다.

```
//지우기 기능
참조 1개
private void button1_Click(object sender, EventArgs e)
{
    //기존 이미지 지우고
    pictureBox1.Image = null;
    OriginalBmp = null;

    //새로 초기화 선언
    listRect = new List<Rectangle>();
    tempRect = new List<Rectangle>();
    listTool = new List<PaintTools>();
    tempTool = new List<PaintTools>();
    pictureBox1.Image = new Bitmap(Application.StartupPath + @"\DefaultBackground.png");

    //Load한 이미지 크기를 PictureBox 컨트롤 크기와 동일하게 맞추어 줍니다.
    pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

    OriginalBmp = (Bitmap)pictureBox1.Image;
    imgRect = new Rectangle(0, 0, pictureBox1.Width, pictureBox1.Height);
}
```

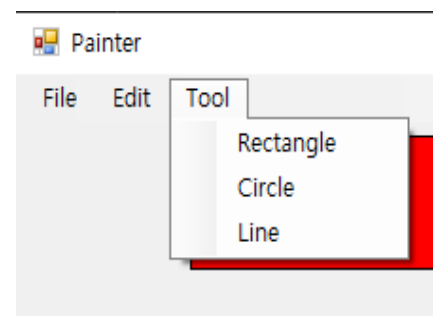
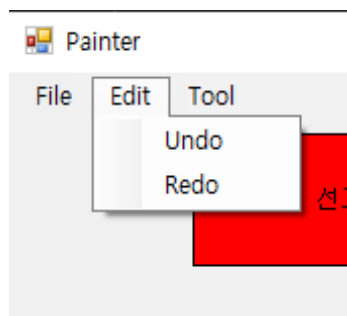
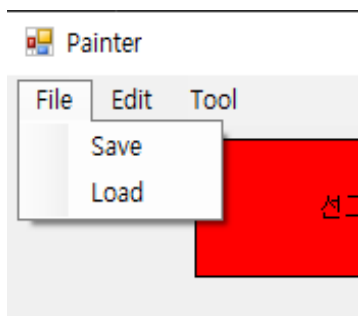
가운데 지우기 버튼은 기존의 흰 배경을 다시 불러오는 기능입니다. 불러오는 동시에 List를 초기화 하고 새로 흰 배경을 입히는 기능입니다.



```
//사용자 색깔 지정
참조 1개
private void button7_Click(object sender, EventArgs e)
{
    ColorDialog dlg = new ColorDialog();

    if (dlg.ShowDialog() == DialogResult.OK)
    {
        pn = new Pen(dlg.Color, width);
    }
}
```

오른쪽 사용자 지정 색 버튼은 색을 새로 입힐 수 있게 ColorDialog을 선언하여 기능을 구현했습니다.



Menustrip 구성은 File, Edit, Tool로 총 3개의 카테고리로 구성했습니다.

```

// 이미지 불러오기
참조 1개
private void loadToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        //이미지 불러오기
        pictureBox1.Load(openFileDialog1.FileName);

        //새로 초기화
        listRect = new List<Rectangle>();
        tempRect = new List<Rectangle>();
        listTool = new List<PaintTools>();
        tempTool = new List<PaintTools>();
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        OriginalBmp = (Bitmap)pictureBox1.Image;
        imgRect = new Rectangle(0, 0, pictureBox1.Width, pictureBox1.Height);
    }
}

// 이미지 저장하기
참조 1개
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult result = saveFileDialog1.ShowDialog();
    if (result == DialogResult.OK)
    {
        pictureBox1.Image.Save(saveFileDialog1.FileName);
    }
}

//파일 save, open 다이얼로그
참조 1개
private void saveFileDialog1_FileOk(object sender, CancelEventArgs e)
{
}

참조 1개
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
}

```

File에서는 파일을 저장하고, 불러오는 Save, Load 기능을 구현했습니다.

Load를 하면 이미지를 null로 초기화하고 List를 새로 선언합니다.

```

//undo
참조 1개
private void undoToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (listRect.Count > 0)
    {
        //Undo 실행 취소
        //맨 마지막 list에 있던 거 temp에 저장하고 마지막 list 삭제
        tempRect.Add(listRect[listRect.Count - 1]);
        listRect.RemoveAt(listRect.Count - 1);
        tempTool.Add(listTool[listTool.Count - 1]);
        listTool.RemoveAt(listTool.Count - 1);
        pictureBox1.Refresh();
        DrawBitmap();
    }
}

//redo
참조 1개
private void redoToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (tempRect.Count > 0)
    {
        //Redo 다시 실행
        //맨 마지막 temp에 있던 거 list에 저장하고 마지막 temp 삭제
        listRect.Add(tempRect[tempRect.Count - 1]);
        tempRect.RemoveAt(tempRect.Count - 1);
        listTool.Add(tempTool[tempTool.Count - 1]);
        tempTool.RemoveAt(tempTool.Count - 1);
        pictureBox1.Refresh();
        DrawBitmap();
    }
}

```

Edit는 그린 그림의 List를 조작하는 Undo와 Redo로 Undo는 그린 그림의 List를 전으로 돌려 Ctrl+z의 기능을 하고 Redo는 반대로 전으로 돌린 List를 다시 원래대로 돌려 놓는 기능을 수행합니다.

```

//menustrip 선, 원, 사각형 그리기 클릭 이벤트
참조 1개
private void rectangleToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolType = PaintTools.DrawRectangle;
}

참조 1개
private void circleToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolType = PaintTools.DrawCircle;
}

참조 1개
private void lineToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolType = PaintTools.DrawLine;
}

```

Tool은 위 3개의 버튼과 같은 기능을 수행하여 선, 원, 사각형을 그리는 기능을 수행합니다.

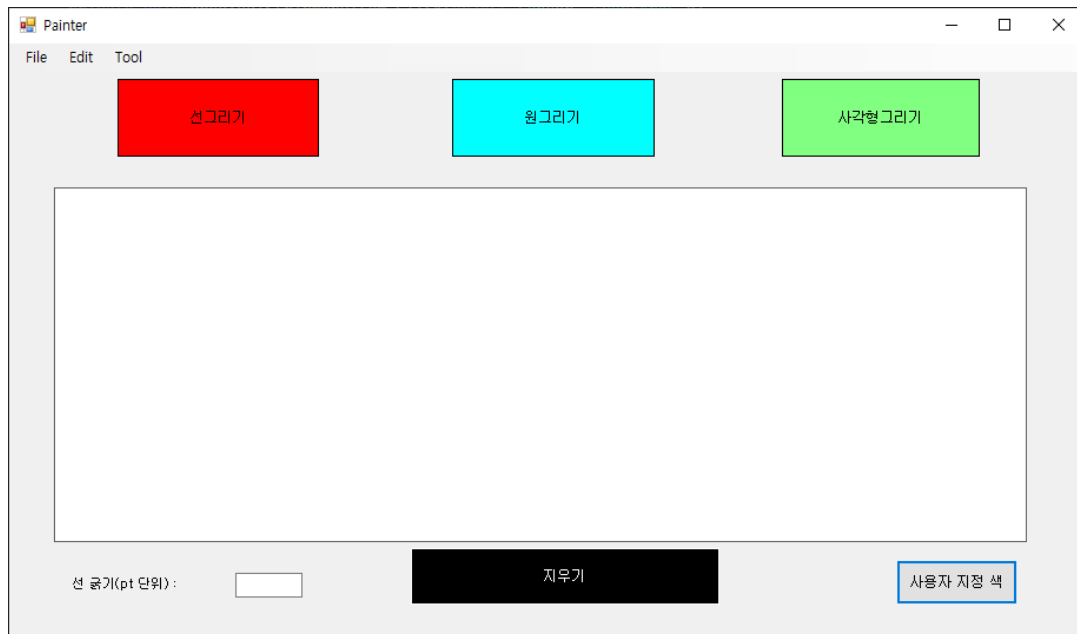
## 문제점 및 고찰

1. 색깔과 선 굵기 구현 중 문제점이 그림을 그렸을 시 한 획을 그리면 다른 선들도 같이 굵기와 색깔이 동시에 변하는 문제를 확인
  - 이에 대한 고찰이 for문으로 굵기와 색깔을 지정하려해서 List에 있는 그림들이 전부 바뀌는 것을 확인했습니다.
2. 이미지를 불러오고 그림을 그리면 그린 선들이 확대되거나 얇게 보이는 현상이 있습니다.
  - 이에 대한 고찰은 pictureBox1에 이미지를 불러오려하니 pixel의 크기를 pictureBox에 맞춰서 그림을 넣어서 그렇습니다.

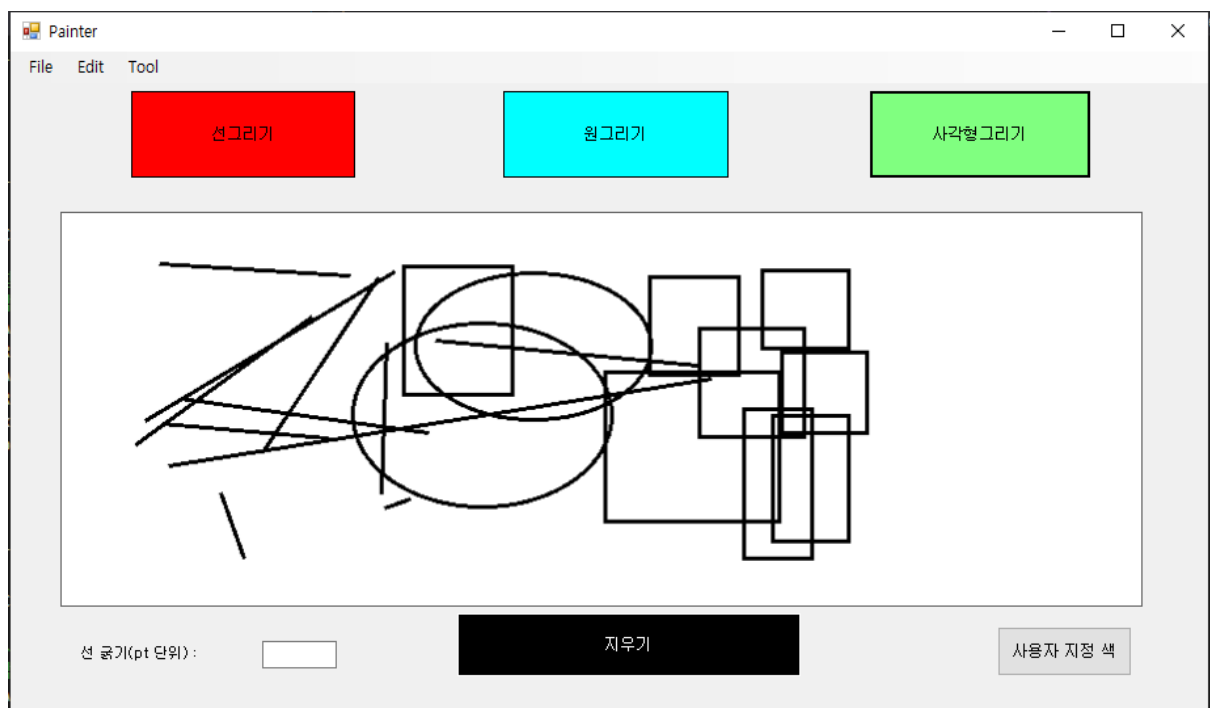
### 3. 실행 화면

#### 시나리오 테스트

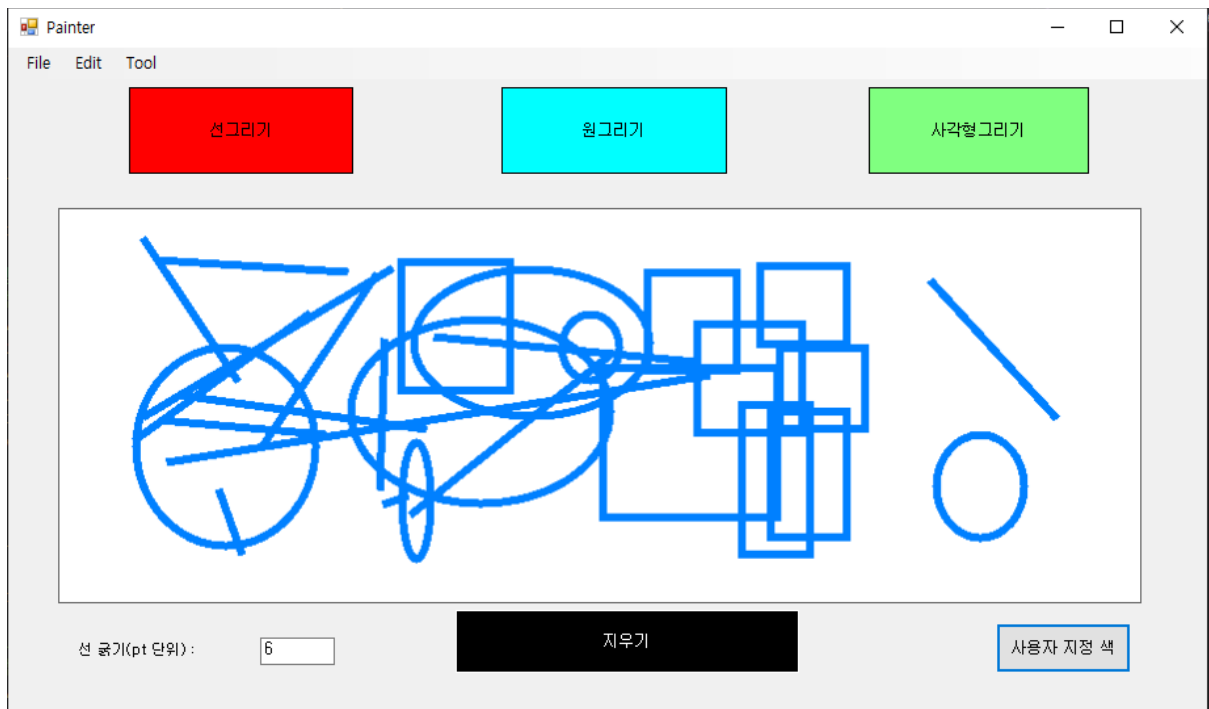
##### - 기본 레이아웃



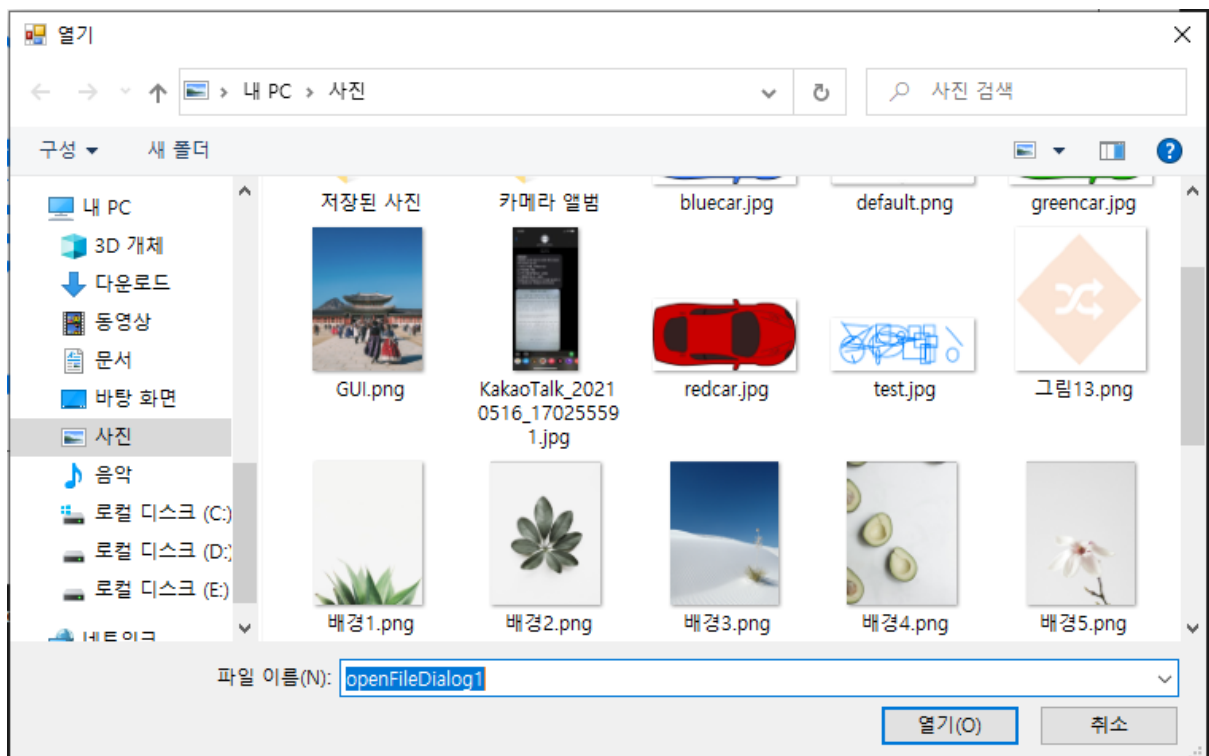
##### - 그리기



## - 굵기, 색 지정



## - 이미지 저장, 불러오기



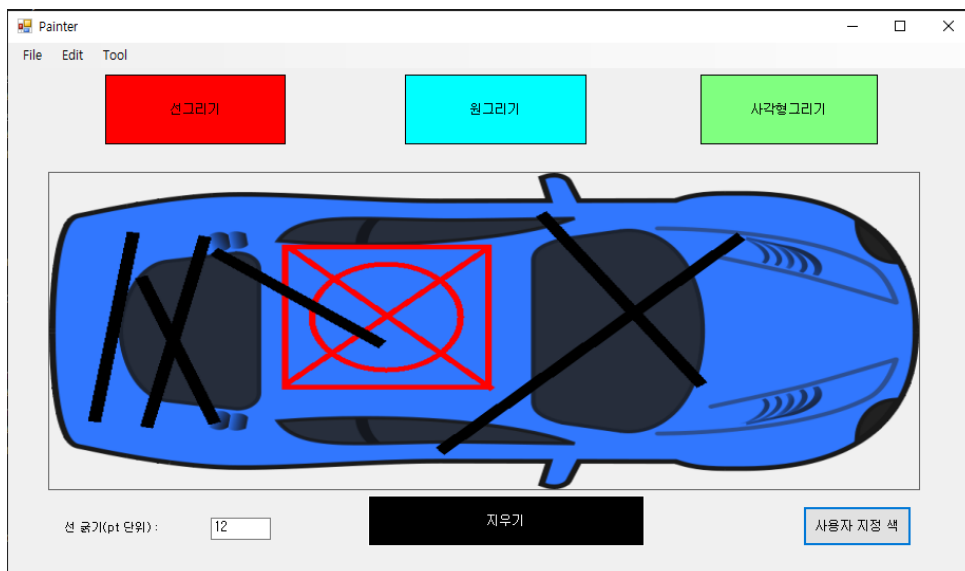
저장된 이미지 확인



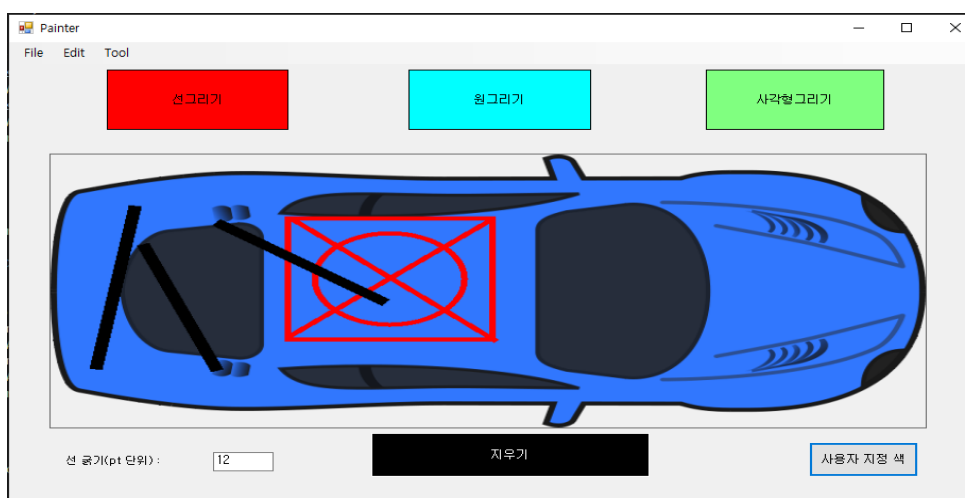


새 이미지에 그림을 그려 저장

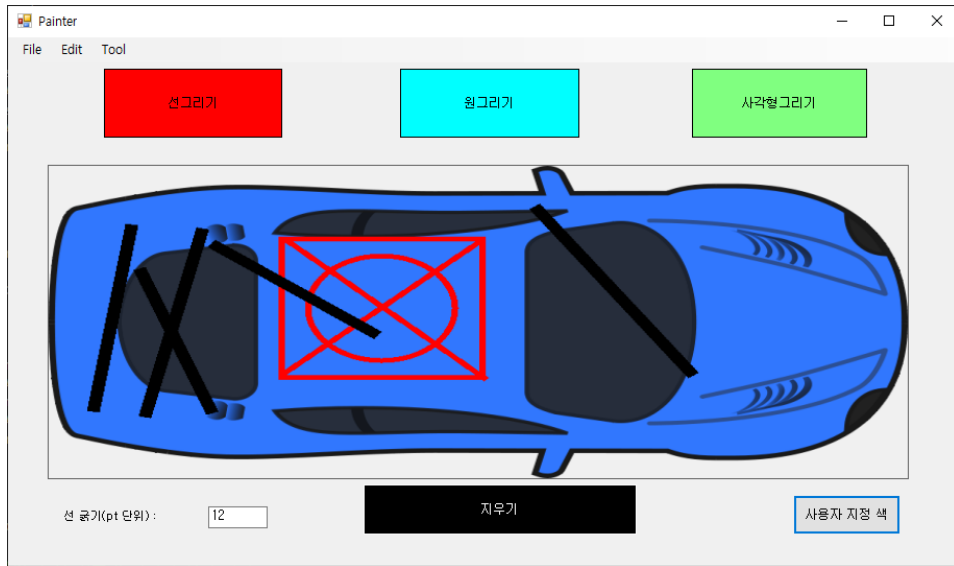
- 저장된 이미지를 불러와 새로 그리기



- Undo와 Redo의 기능



4번의 Undo로 그린 내용 되돌리기



2번의 Redo로 그린 내용을 다시 불러오기

## 4. 개발 환경

### - 프레임워크 버전

The screenshot shows the 'Project Properties' dialog box in Visual Studio for a project named 'Painter\_Project'. The 'General' tab is selected. At the top, '구성(C):' (Configuration) and '플랫폼(M):' (Platform) are both set to 'N/A'. The '어셈블리 이름(N):' (Assembly Name) is 'Painter\_Project', and the '기본 네임스페이스(L):' (Default Namespace) is also 'Painter\_Project'. Under '대상 프레임워크(G):' (Target Framework), '.NET Framework 4.7.2' is selected. Under '출력 형식(U):' (Output Type), 'Windows 애플리케이션' (Windows Application) is selected. The checkbox '바인딩 리디렉션 자동 생성(A)' (Automatically generate binding redirects) is checked. The '시작 개체(O):' (Startup Object) is set to '(설정 안 함)' (Not set). A button labeled '어셈블리 정보(I)...' (Assembly Information...) is visible on the right. The bottom of the dialog shows the text '리소스' (Resources).

.NET Framework 4.7.2로 구현

### - 운영체제

Windows 10